



UNIVERSIDADE FEDERAL DE PELOTAS
CENTRO DE ARTES
COLEGIADO DOS CURSOS DE CINEMA

JEFFERSON E SILVA NASCIMENTO

FLUXO DE ANIMAÇÃO MULTITÉCNICA COM BLENDER

Pelotas/RS

2019

JEFFERSON E SILVA NASCIMENTO

FLUXO DE ANIMAÇÃO MULTITÉCNICA COM BLENDER

Artigo científico apresentado como requisito parcial para a obtenção do grau de Bacharel em (Cinema e Audiovisual ou Cinema de Animação) no Centro de Artes da Universidade Federal de Pelotas.

Orientador: Prof. Maurício Gonçalves

Pelotas/RS

2019

RESUMO

O presente artigo tem como propósito detalhar o fluxo de produção de séries de animação multitécnica com a utilização da ferramenta gratuita e de código aberto Blender 3D. Tal estudo tem foco na análise crítica de técnicas e procedimentos de animação, observados através da produção prática do autor ao longo do curso e focando no curta de conclusão de curso Ian. Busca-se, também, estabelecer um panorama do desenvolvimento de programas de código aberto para animação 2D e como tais soluções são desenvolvidas pela comunidade de usuários. Como procedimento metodológico, a pesquisa adota um caminho empírico de observação e análise, apoiado por referenciais teóricos relacionados aos temas de imagens técnicas, cibercultura e multidisciplinaridade.

Palavras-chave: fluxo de trabalho, Blender 3d, animação 2d, mercado de séries de animação

Sumário

1. Introdução	5
2. Histórico do <i>software</i> Livre de Animação 2d	7
3. <i>software</i> , artista e comunidades	13
4. O Fluxo de Trabalho	14
5. <i>Storyboard</i> e <i>Animatic</i>	15
6. Personagens 2d	17
7. Personagem e animação 3d	22
8. Partículas	25
9. Animação Paramétrica	26
10. Render	27
11. Considerações finais	29
Abstract	34
Referências Bibliográficas	35
Filmografia	36
Índice de Figuras	37
Anexos	38

1 Introdução

Mesmo antes do curso eu tinha interesses em *software* livre, durante o curso, o estudo e interesse se aprofundaram. Aprendi bastante sobre diversos *softwares* e fluxos de trabalho, a partir desse conhecimento propus oficinas e aulas onde compartilhava do conhecimento com os colegas e professores. Durante toda minha história no curso de cinema de animação procurei introduzir o uso dessas ferramentas. É interessante notar que duas delas tiveram uma grande adoção, krita, *software* de pintura digital e animação tradicional, e Blender 3d, o *software* alvo do artigo.

O *software* assim como o *hardware* e seus periféricos, elementos físicos necessários para o seu funcionamento, tornaram-se a ferramenta padrão do animador atual. Cada recurso de *software* substitui e corresponde a uma etapa da produção de uma animação. Um *software* de desenho substitui o papel, pincéis e tintas, um de animação substitui mesas de luz, câmeras e acetatos, um de edição substitui a moviola e assim por diante. Dessa forma, estamos sujeitos aos usos dessas “Caixas Pretas de Flusser”. Conceito em que o filósofo tcheco, naturalizado brasileiro, Vilém Flusser (1985) fala sobre a manipulação do que ele chama de “aparelho”, tais aparelhos geram imagens através de intrincados sistemas (no nosso caso, o *software* de computação gráfica), cujo manipulador desconhece e nem precisa conhecer. Esses sistemas geram o que chamamos de imagens técnicas.

Por ser uma tarefa complicada e cheia de etapas, para que uma animação seja desenvolvida, desde o roteiro e o *storyboard*, conceitos, áudio, animação até a finalização e a montagem, são necessárias ferramentas compostas de *hardware* e *software*, estes muitas vezes especializados. Essa especialização vem com custos, muitas vezes elevados e, para se manter um estúdio produtivo, somam-se ainda outros custos como a mão de obra e encargos diversos. No âmbito do artigo, olhamos para os gastos e problemas gerados pelo uso de *software* proprietário. Com a intenção de apontar soluções para a redução desses custos, optou-se por uma abordagem envolvendo alternativas de *software* livre.

O objeto de pesquisa é o curta Ian (2019), fruto de trabalho em equipe e curta de graduação do autor. Analisaremos e descreveremos as etapas em que foram usados as ferramentas do Blender e as dificuldades da implementação do mesmo no nosso fluxo de trabalho. Temos como objetivo delinear uma forma de guia inicial do uso do mesmo na criação de animações de técnica mista e

analisar de forma crítica o histórico e a formação do *software* livre Blender e sua comunidade utilizando de teoria de imagens técnicas, cibercultura e transdisciplinaridade.

Ian foi um curta produzido com o intuito de ser um piloto de série, toda sua elaboração desde o início teve esse enfoque, fizemos pesquisas de mercado e nos adaptamos ao que achamos que seria algo interessante de se criar. O personagem principal do curta é Ian, um menino que tem por volta dos 6 anos de idade e muitos brinquedos. Em seu quarto ele e seus dois amigos, Aruna, um urso de pelúcia lutador de sumô, e Beto, um boneco de um gatinho com óculos e gravata, se aventuram em cenários imaginários relacionados com as situações reais que o menino está vivendo. No primeiro episódio Ian chega em seu quarto e sua mãe fala para ele ter cuidado com o chão limpo do quarto, Ian se imagina em uma caverna cheia de lava e pula de pedra em pedra para evitar ser queimado, enquanto isso ajuda seu amigo Aruna, que está preso em um local que o chão está cedendo. No final do episódio a mãe de Ian pergunta o que está acontecendo e o garoto sai do mundo imaginário e percebe seu quarto todo bagunçado.

A análise do curta será feita a partir da separação de etapas que tiveram o Blender como ferramenta principal. Observamos como as etapas foram realizadas, suas vantagens e complicações e trouxemos uma reflexão sobre melhorias possíveis. As etapas observadas são *storyboard* e *animatic*, personagens 2d, personagem e animação 3d, partículas, animação paramétrica e render.

O estudo tem como base a teoria das imagens técnicas, onde observamos a categorização de aparelho e a interação operador/aparelho e como isso se conecta ao *software* livre e seu usuário. A partir dessa reflexão podemos adentrar ao campo da cibercultura observando o fenômeno através da comunidade em torno desses *softwares* e como ela se desenvolve. E os estudos de transdisciplinaridade nos ajudam a entender como a forma de pensar transdisciplinar ajuda a moldar o *software* e o operador com uma reflexão sobre os problemas que isso gera para o operador.

Também traremos uma breve análise histórica do cenário de *software* livre voltado para animação 2d, para isso recorreremos a documentação online a partir de 2016. Como um último objetivo temos a criação de um infográfico que possa ajudar os alunos do curso a entender as possibilidades e como seria o possível fluxo de trabalho.

O *software* livre nos traz muita liberdade, para além do que um *software* gratuito de código aberto. Com ele podemos, além de adaptar nosso fluxo de trabalho, adaptá-lo ao nosso fluxo, claro, com os conhecimentos certos. Isso sem quebrar licenças e acabar prejudicando a nós, criadores. Ora, qual a importância da modificação desses *softwares* já tão especializados? Olhando para o mercado e pensando nos grandes estúdios se observa algo, muitos deles desenvolveram suas

próprias ferramentas internas, pois assim, provavelmente, não infringiriam nenhum tipo de licença e tem a liberdade para adaptar a ferramenta para qualquer necessidade.

Fazer um *software* do zero se torna, entretanto, algo muito custoso e somente grandes estúdios ou estúdios que comecem com pessoas especializadas nessa área conseguiriam fazer algo nesse sentido. Como exemplo temos a Pixar, cujas soluções foram e são desenvolvidas internamente. Assim, entra a importância da nossa escolha, pois pensando em um futuro onde possamos ter um estúdio sob nosso comando é bem vinda a possibilidade da alteração da ferramenta conforme necessário. Em uma situação hipotética podemos contratar uma pessoa especializada para desenvolver soluções internas que não infringem o uso do *software*, podendo nos poupar tempo e dinheiro.

Essas implicações podem ser observadas em estúdios que já adotam o Blender como ferramenta constante e principal em seu fluxo de trabalho. O filme *Next Gen* (2018) foi realizado pelo estúdio *Tangent Animation*, em uma palestra de 2018, *Blender and Next Gen: a Netflix Original*, na *Blender Conference*¹ o estúdio demonstrou entre técnicas de produção os seus melhoramentos feitos ao código fonte do Blender 3D. Essas melhorias foram reincorporadas ao Blender logo após avaliação da *Blender Foundation* e sua comunidade.

2 Histórico do Software Livre de Animação 2d

No modo de entender desta pesquisa, até meados de 2016 não existiam boas alternativas para animação 2d no campo do *software* livre. Na época apenas identifiquei uma possibilidade que bateria de frente com o fluxo de trabalho do *Adobe After Effects*: o Blender 3D. O fluxo se assemelha devido ao fato de ambos *softwares*, na época, não contarem com formas de ilustração integradas. Ambos tinham que ter as ilustrações importadas e preparadas para etapas posteriores.

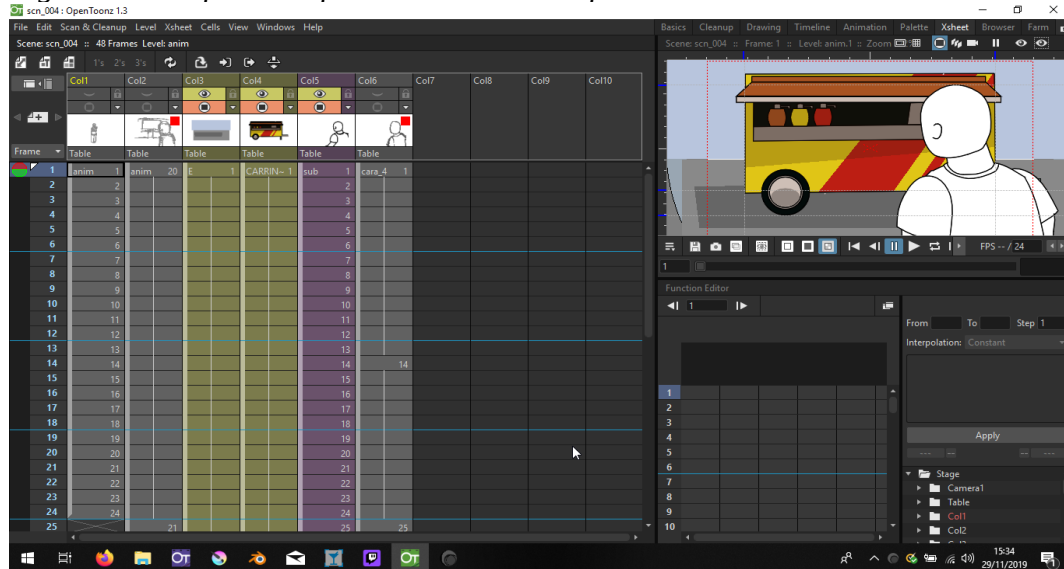
Em 2016 surge uma notícia na comunidade *opensource*², o *software* Toonz teve seu código aberto e foi liberado sem custos para todos. Agora chamado de Opentoonz (FIGURA 1) o *software* carrega uma história na produção de longas e curtas pelo mundo. Como visto em matéria de Amidi (2016) no site Cartoon Brew é interessante ressaltar que o Opentoonz é a versão do Toonz personalizada pelo estúdio Ghibli. É possível até mesmo observar seu uso no documentário *The*

1 Conferência anual realizada pela *Blender Foundation* para discussões em torno dos usos e avanços do Blender

2 O termo significa código aberto, um *software opensource* tem seu código disponível para visualização e possível modificação.

Kingdom of Dreams and Madness (2013), sobre o estúdio. O autor com uma equipe de animação utilizaram o *software* para produção de um curta, *Jerry, O Coala* (2016) (FIGURA 2) e o autor continuou a utilizar o *software* para outros projetos.

Figura 1: Interface do opentoonz com um arquivo do autor aberto.



fonte: acervo pessoal.

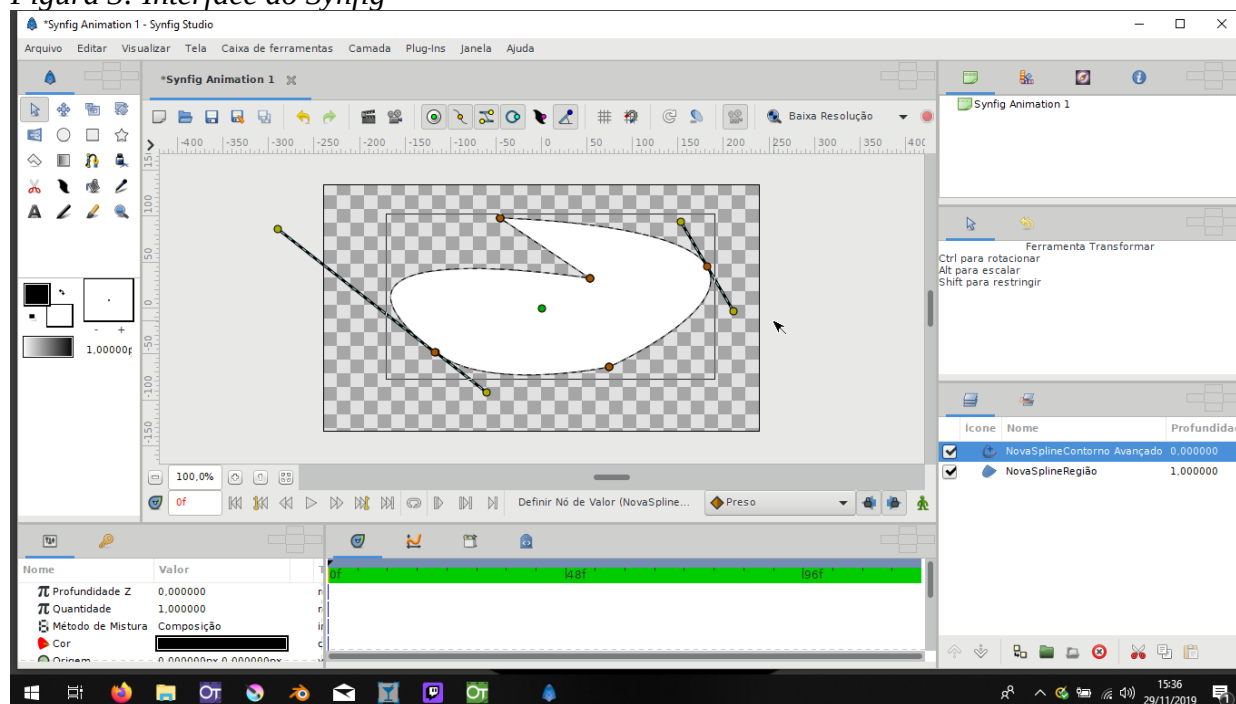
Figura 2: Quadro do curta *Jerry, O coala*.



fonte: *Jerry, O Coala* (2016)

No mesmo ano a comunidade *opensource* teve outras notícias para animação 2D, um projeto parado chamado Synfig (FIGURA 3) foi retomado, um *software* de animação vetorial por interpolação. O *software* de pintura digital Krita inaugurou um módulo de animação quadro a quadro. O Blender 3D ganhou uma extensão³ para facilitar seu uso como *software* de animação 2d, feito por um membro da comunidade interessado na integração entre o Blender e uma game engine de nome Godot. Andres Esau criou e distribuiu o *Cut Out Animation Tools*. Mistura entre ferramentas novas e organização da interface que facilita o desenvolvimento de 2d baseado em ilustrações produzidas previamente no Blender. A extensão em questão foi utilizada pelo autor com outra equipe para produção do videoclipe 313 Experimentos (2018) (FIGURA 4) da banda laboratório suicida.

Figura 3: Interface do Synfig



fonte: acervo pessoal.

³ Extensão, segundo a organização mozilla é um “conjunto de bits de código que modificam a funcionalidade do navegador.” Em vista de que o navegador é um *software* essa definição pode ser estendida. O usuário pode se deparar com outros nomes, mais utilizados notadamente em inglês, *add-on*, *addon*, *plug-in* ou *plugin*.

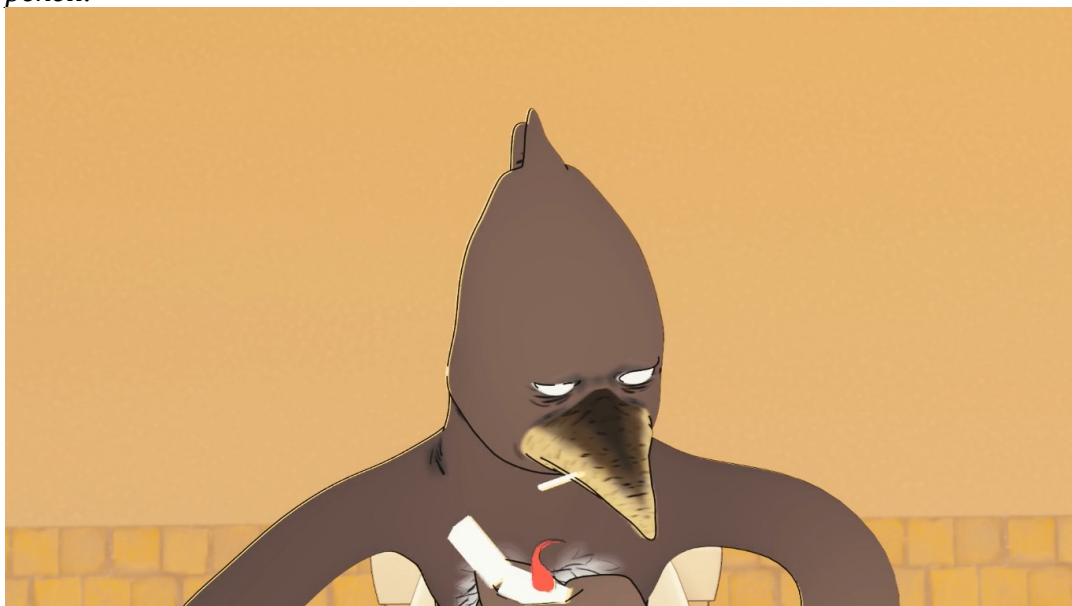
Figura 4: Cena do videoclipe 313 experimentos de Gerson na Interface do Blender 2.79b com a extensão COA Tools.



fonte: acervo pessoal.

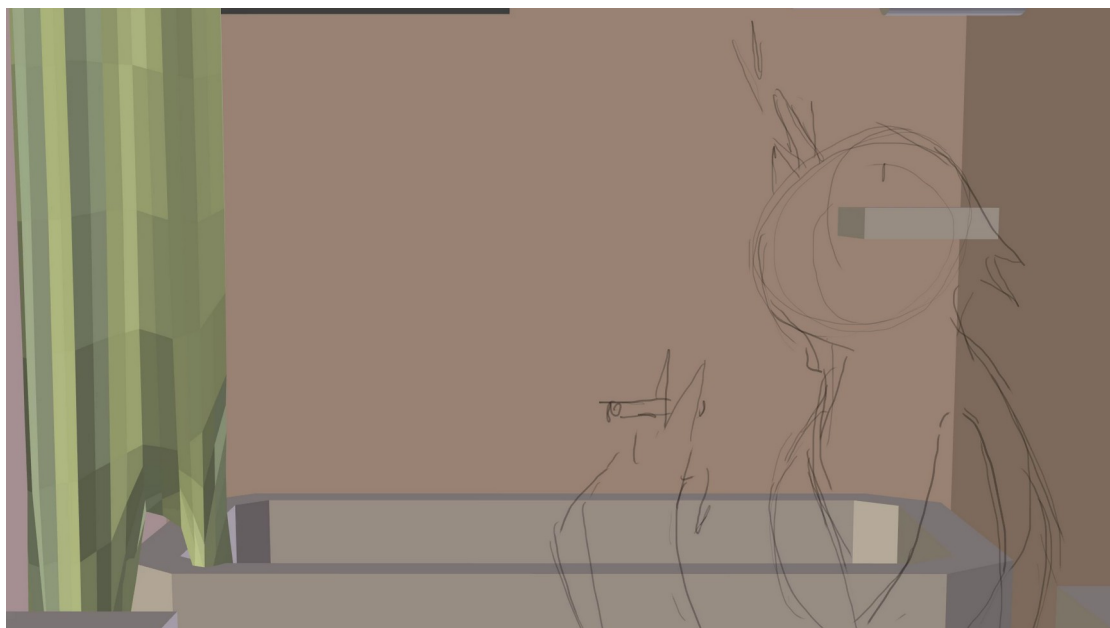
No período de 2016-2018 houve outros curtas e experimentos usando o Blender 3D e uma ferramenta, a princípio para anotações, no contexto de animação 2d. O *grease pencil*, ferramenta com a qual é possível desenhar, animar e dar saída em animações quadro a quadro e interpolações simples. O autor usou estas ferramentas para o *storyboard* do curta Jerry, o Coala(2016) e para *storyboard*, efeitos e animação 2d integradas no ambiente 3d do curta A Inveja de Cléber e um Cigarro no Banheiro (2017) (FIGURA 5 e 6).

Figura 5: Quadro do curta A Inveja de Cléber e um cigarro no banheiro. O detalhe do fogo do isqueiro é feito em 2d em espaço tridimensional com *grease pencil*.



fonte: A inveja de Cléber e um cigarro no banheiro (2017)

Figura 6: Quadro do animatic do curta, desenhando em espaço tridimensional.



fonte: acervo pessoal.

É importante que saibamos que em 2018 começa o desenvolvimento da versão 2.8 do Blender 3d, nele a ferramenta *grease pencil* ganha destaque. Um curta foi feito para o aprimoramento dos fluxos de trabalho e da ferramenta, *Hero* (2018), forma de trabalhar adotada pela *Blender Foundation*⁴ desde *Elephant Dreams*(2006), onde curtas são produzidos com o intuito de melhorar o *software*.

É nesse contexto que surge a ideia de nosso filme de conclusão de curso, *Ian*, um produto seriado. Nossa escolha pelo Blender como opção central de animação vem não só do histórico do autor com o *software* ou da equipe em que quase todos já tiveram experiência com o *software* devido a participar de projetos com o autor, mas devido a nossa estética visual baseada em *Incrível Mundo de Gumball* (2018). Isso significaria misturar diversas técnicas, no nosso caso, 2d de recortes, quadro a quadro, 3d, efeitos visuais, pós produção, iluminação e técnicas ainda não desenvolvidas, mas possíveis, como 2d em espaço tridimensional. O *software* acabou sendo uma escolha natural, já que assim não espalharíamos a produção em diversas partes introduzindo muitas variáveis técnicas tais como integração de diferentes tipos de arquivos por diversos *softwares*, por exemplo a importação de modelagem 3d de um personagem em um ambiente de *software*

especializado em 2d. Também surgiu como preocupação da equipe o uso do *software* pirata caso o projeto fosse levado adiante pela mesma equipe, esta preocupação contribuiu para a escolha do Blender como central de produção.

Outro ponto é a recente criação da *Academy software Foundation*. De acordo com a página sobre a fundação foi “o resultado de uma pesquisa de dois anos feita pelo conselho de Ciência e Tecnologia da *Academy of Motion Picture Arts & Sciences (AMPAS)* (FIGURA 8) sobre o uso de *softwares* open source”. (Tradução livre) Segundo o conselho 84% dos produtores de conteúdo audiovisual estavam utilizando *softwares* e soluções open source na linha de produção. Isso impeliu a academia a criar essa fundação com o intuito de organizar a produção e compartilhamento de informações. A lista de soluções abraçadas pela fundação é grande e entre elas está o Blender 3d.

Figura 7: Cartaz de divulgação dos projetos opensource apoiados pela AMPAS.

OPEN SOURCE IN THE MOTION PICTURE INDUSTRY

Open source software is at the core of the creative universe, helping incredible worlds and stories come to life on the big and small screen. And there are still many frontiers to explore.

The Academy of Motion Picture Arts and Sciences - in partnership with The Linux Foundation - has formed the **Academy Software Foundation** to support the core infrastructure of the motion picture industry and enable even greater innovation.

We salute the hard work from developers and businesses that has allowed the industry to advance in exciting ways, and we are honored to be part of the future of open source and creative entertainment as we help unify these efforts.

These are just some of the amazing projects developed by the motion picture industry

ACES	Houdini Ocean Toolkit	Partio
Afanasy	Hugin	pfstools
Affogato	Imagemagick	Pimath
Alembic	Jupiter Skin	Pixie
ANNN	JupiterFileCache	Processing
Aqsis Renderer	LiquidMaya	Ptex
Arsenal Suite	Maya Reticle	PyMEL
Blender	Open Cinema Tools	PyODE
Bullet Physics	Open Color I/O	pyShake
CGAL	Open Dynamics Engine	pyString
CgKit	Open Shading Language	Rez
Cinepaint	OpenSubdiv	RmanConnect
Cortex	OpenVDB	rmanPtcSOP
DJV Imaging	OpenBatchIO	Scala Migrations
dnPtcViewerNode	OpenDCP	SeExpr
DrQueue	OpenDisplayCalib	Sun Grid Engine
Dynamica	OpenEXR	TACTIC
Earth	OpenFX	tracksperanto
Field3D	OpenImageIO	Universal Scene Description
Gaffer	OpenPipeline	vfxgal
GIMP	OpenRenderManagement	Wavelet Turbulence
GTO	OpenSequencePlayer	for Fluid Simulation
GTS	OpenTimelineIO	

Project list from <http://opensourcesfx.org/>

fonte: <http://www.aswf.io/>

3 *software*, artista e comunidades

O *software* de computação gráfica utilizado pelo artista para gerar suas imagens, em conjunto com o computador e periféricos é uma caixa preta. Flusser (1985) escreve o que é uma caixa preta, um aparelho. Segundo o autor o aparelho surge como objeto pós industrial, numa lógica em que se usa informação para produzir informação. O operador deste aparelho tem que saber como alimentá-lo para ter o que receber desse aparelho, pois o que ocorre entre esse processo é obscurecido, daí o termo caixa preta. O computador em si, *hardware*, vai ser uma caixa preta que gera condições de automatização e o *software* de computação gráfica, *software*, é a caixa preta que te permite gerar informações através de automatização. Em comum, alguém programou esses objetos para realizar funções e o operador, no caso, artista, vai se adaptar ao seu funcionamento e gerar as imagens técnicas.

Em questão ao operador em O Mundo Codificado, Flusser(2008) indaga sobre como o operador perde liberdade dentro da ilusão de liberdades que os aparelhos trazem. É importante portanto refletir, como artista, o que se faz ao se utilizar tais ferramentas. Ao lermos Não-coisa (1) e (2) é possível entender a preocupação do autor com as escolhas singulares que foram codificadas nesses programas e a relação com a ilusão passada por essas escolhas, pois as respostas que deveríamos alcançar ao nos propormos a atividade através deles já estão codificadas. A reflexão do artista então se torna importante no momento da operação, do jogar, de entender que o programa traz respostas prontas, mas que devemos sim utilizá-las ao nosso favor para gerar nossas respostas e não deixar-se automatizar.

É interessante observar as definições de *software* do ponto de vista da computação. *Software* “é um conjunto de instruções desenhado para realizar uma série de processamentos de dados introduzidos e produzir certos resultados. Ele é escrito para lidar com sistemas de Entrada-Processamento-Saída alcançando objetivos pré determinados” Jawadekar (2004) (Tradução livre). Essa definição não escapa do que Flusser(1985) discute, em que os aparelhos são caixas pretas em que se entram coisas, algo acontece e saem coisas novas. Levy(2009) não se estende ao escrever que *softwares* são feitos para automatizar funções, e que diferentes camadas de *software* podem se conectar antes do acesso desse *software* ao *hardware* para execução das tarefas.

Em Flusser (1985) há a discussão de como o poder se desloca para a mão do programador desses aparelhos. Este que comanda o que vai ou não ser utilizável para o operador. É interessante o que podemos imaginar que o *software* livre traz para esta situação, a abertura da caixa preta, pelo menos parcial. Essa abertura parcial se dá pelo fato de que o operador disposto a adentrar e se

empoderar da programação do próprio *software* terá a barreira do conhecimento sobre a linguagem a qual o *software* foi programado. Levy (2009) discorre sucintamente sobre o assunto, os programas tem que ser feitos de alguma forma e esta forma são as linguagens de programação, “códigos especializados para escrever funções para processadores de computadores.

Em O Mundo Codificado, Flusser(2008) traz essa reflexão mais aprofundada em não-coisa (2), essa barreira imposta seria um metaprograma, cujo programador deve ser operador para criar assim o programa ao qual outro operador opera. Levy (2009) adiciona pontos sobre como essa programação funciona hoje ao trazer à tona a descentralização da criação, segundo o autor, típica das artes do virtual. Apesar da programação de um *software* não pertencer ao campo artístico se observa a descentralização do desenvolvimento do mesmo, com múltiplos programadores (aquele que constrói o programa através de programação) em torno do mundo conectados à internet resolvendo problemas ou implementando novidades. Artistas ajudam com sugestões, testes ou se convertem em híbridos artistas-programadores. Tudo isso volta pra comunidade e deságua em conferências e encontros.

Ao falarmos de *software* livre é importante entendermos melhor do que se trata. Quando observamos *softwares* que adotam código aberto as coisas podem se complicar um pouco, segundo Stallman (2019), todo *software* livre é de código aberto, mas nem todo *software* de código aberto é livre. Isso se deve ao fato de que as licenças de *software* livre carregam também em sua essência direitos de liberdade ao operador, como modificação e distribuição do código fonte. A recíproca não é verdadeira para o *software* que carrega uma licença de código aberto, cuja única garantia é acesso ao código fonte do *software*. Stallman faz a seguinte comparação, o *software* livre é grátis, mas não como em cerveja grátis, mas como em liberdade de expressão. Apesar de confusa, a analogia nos permite pensar como o *software* livre opera, não apenas é algo dado em que nossa relação é apenas de consumir, mas sim de ouvir, ser ouvido, agregar e se permitir modificar.

4 O Fluxo de Trabalho

Na produção de Ian, apesar da intenção inicial ser o uso do Blender como ferramenta principal decidimos como equipe que se isso fosse um entrave aos participantes do grupo, estes poderiam utilizar a ferramenta com a qual estivessem mais acostumados para evitar atrasos no ritmo de produção. Relatarei aqui quais partes do filme tiveram sua produção focada no uso do Blender, quais poderiam utilizar, mas não utilizaram e problemas que enfrentamos.

O principal motivo do uso do Blender foi a direção de arte escolhida para seguirmos, algo mais solto, onde poderíamos encaixar e misturar estilos e técnicas, logo um *software* que nos trouxesse essa integração de forma simples seria a melhor escolha. A natureza multidisciplinar do Blender então nos conveio, nele podemos fazer animações 2d em recortes ou quadro a quadro, podemos inserir elementos tridimensionais diversos e ter sua saída em inúmeros estilos, do realismo à estilização mais extrema, podemos brincar com partículas, tecido, física realista, inserir trechos em vídeo e fazer detecção de movimentos de câmeras. Tudo isso nos levou a essa escolha, para que em futuros episódios exista a liberdade total de criação por parte da equipe sem nos preocuparmos se podemos ou não realizar o episódio tecnicamente.

No episódio piloto, “chão de lava”, utilizamos das capacidades de 2d para animação de recortes, *storyboard* e *animatic*, física de partículas, física de corpos rígidos, modelagem tridimensional, animações paramétricas, animação 3d e *render* realista e não realista.

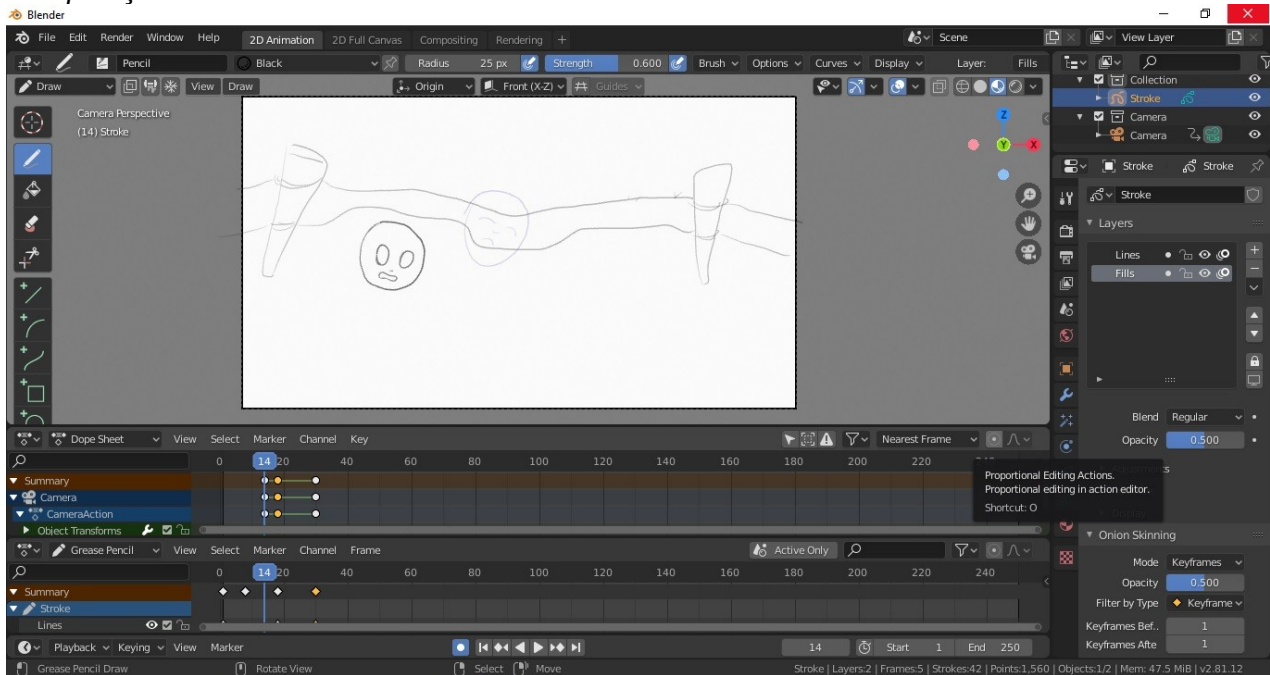
5 *storyboard* e *Animatic*

Desenhar o *storyboard*⁵ diretamente no Blender é algo simples. Como citado anteriormente o autor já o fez em diversas situações, em A Inveja de Cléber e Um Cigarro no Banheiro (2017) foi utilizado de ambiente tridimensional com posicionamento da câmera nos locais para o desenho ser o mais próximo do produto final. Em Ian não foram utilizadas ferramentas tão sofisticadas, apenas desenhos separados em camadas e uma câmera para saída. O mais interessante do fluxo de trabalho para *storyboards* é que devido à possibilidade de importação de áudio para uma linha de tempo por causa das capacidades de edição de vídeo é possível a criação de um *animatic* diretamente na ferramenta. Ian teve seu *animatic*⁶ totalmente construído dentro do Blender se aproveitando dos elementos já desenhados de seu *storyboard*. Nessa abordagem também se aproveitam as possibilidades de animação de elementos e da câmera sempre que necessário.

5 *Storyboard* é o planejamento da estória em sequência de imagens com indicações de movimento.

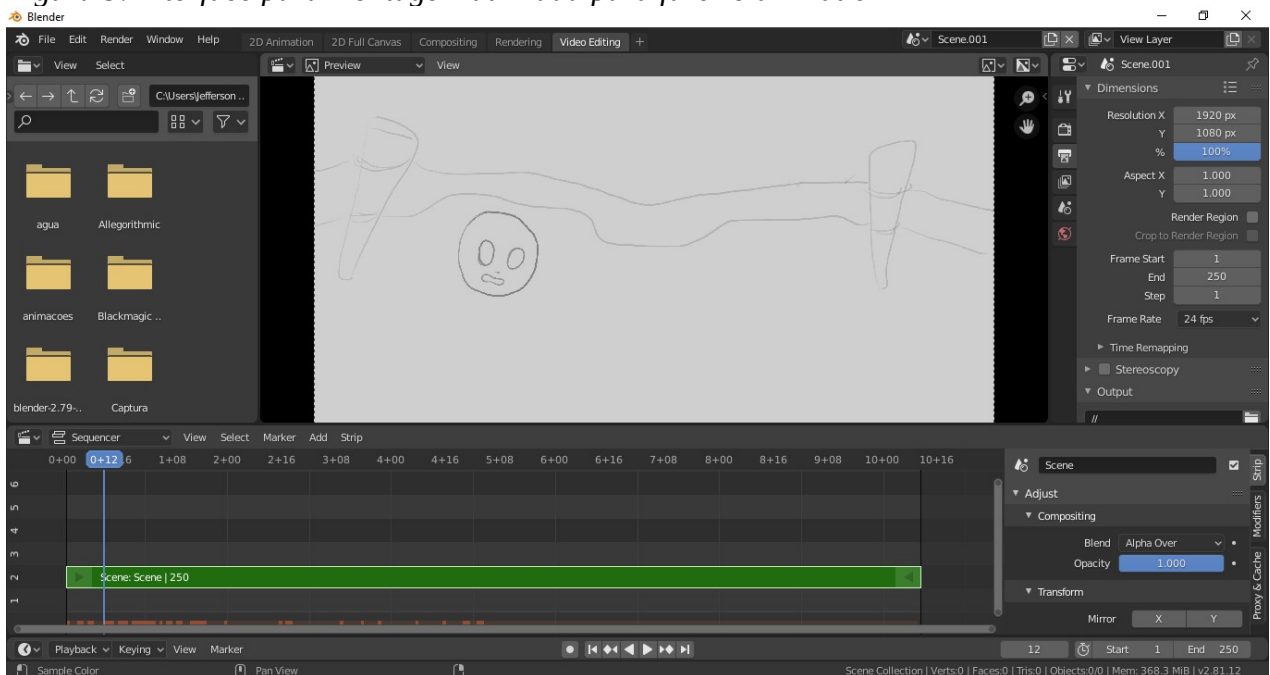
6 *Animatic* é o planejamento do timing do filme utilizando as imagens do storyboard e áudio.

Figura 8: Interface montada para desenho e manipulação de keyframes de desenho e interpolação.



fonte: acervo pessoal, 2019.

Figura 9: Interface para montagem utilizada para fazer o animatic



fonte: acervo pessoal, 2019.

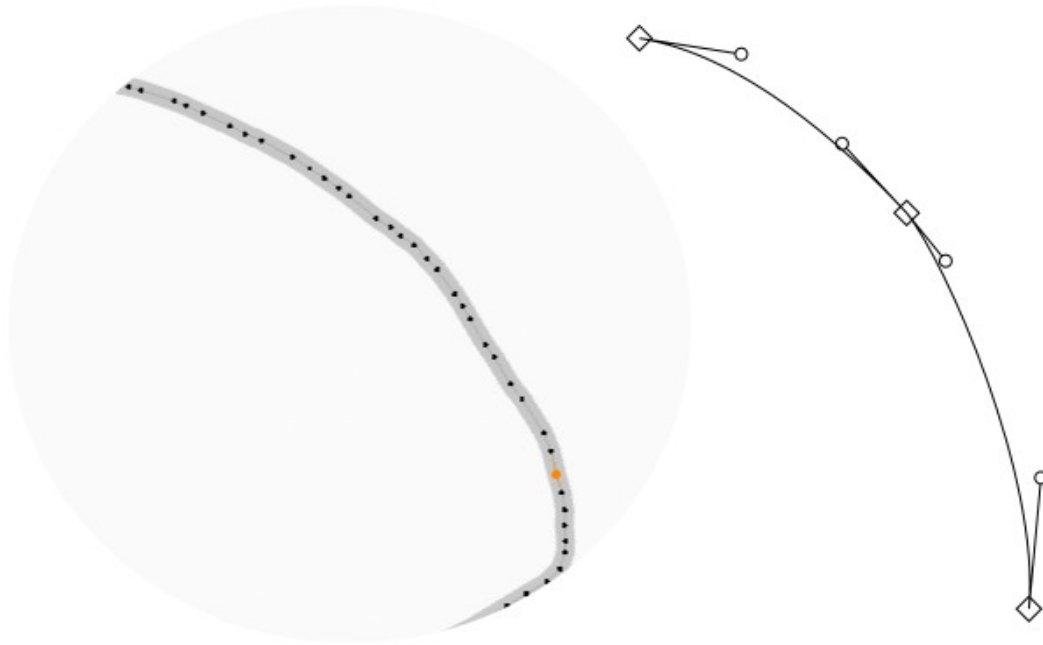
O único contratempo nessa abordagem é a separação do local de edição de quadros chave na interface. Quadros gerados pelos desenhos bidimensionais são separados de quadros de interpolação e separados da montagem de áudio. Essa separação de interface pode interferir um pouco no fluxo de trabalho. A solução mais simples encontrada pelo autor é a personalização da

interface, devido a natureza personalizável do Blender desenvolvemos uma interface ideal para a criação de *storyboards* (FIGURA 9) e usamos a de montagem que já vem no Blender 3d para os animatics (FIGURA 10). Nesta solução a tela é dividida em espaços para gerenciamento de camadas, linha do tempo para quadros chave bidimensionais, de interpolação e edição de áudio e uma área de desenho.

6 Personagens 2d

Antes de qualquer descrição é importante saber que o desenho bidimensional no Blender é feito a partir de vetores, objetos matemáticos que definem pontos e linhas aos quais o computador traduz para a tela em pixels. Vetores têm propriedades interessantes ao processo de criação de uma série, como a possibilidade de aproximação do desenho infinita. Tal recurso diminui a necessidade de diferentes desenhos em diferentes resoluções. O *grease pencil* usa um tipo de vetor feito de diversos segmentos retos interligados pelos seus pontos, diferente do que se vê normalmente em *softwares* vetoriais, as curvas definidas por dois pontos e alças, nessa abordagem os pontos definem início e fim do segmento e as alças definem como a curva se comporta (FIGURA 11). Nesse sentido se aproximando mais de um objeto tridimensional e seus vértices. Para contrapor a possibilidade de controle fino perdida nessa implementação existe um modo de edição em que cada ponto ou muitos pontos ao mesmo tempo podem ser movimentados de diversas maneiras, e um modo de escultura onde esse desenho pode ser esculpido com pincéis que permitem suavização, movimentos entre outras operações. O outro modo que contorna essa limitação é o uso dos modificadores. Um paradigma do uso do Blender, modificadores realizam operações não destrutivas em seus objetos, como suavização, replicação, mudança de cores e no caso do *grease pencil* efeitos como sombras e iluminação. Para maior controle esses modificadores podem ter seus efeitos limitados a grupos de vértices específicos e a potência dessas modificações através dos pesos dados a cada ponto.

Figura 10: Na esquerda o vetor gerado pelo grease pencil e na direita um vetor gerado em um software de edição de vetores comum.



fonte: Acrevo pessoal, (2019).

A técnica de animação de recortes foi utilizada em dois dos três personagens, Ian e Aruna (FIGURA 12). Ambos foram ilustrados e tiveram rigging e animação feitos diretamente no Blender. O personagem Ian tem ilustração mais simples com estilo visual chapado e linhas sem variações. Aruna apresenta leve textura fotográfica e linhas que tentam emular o traço de um pincel. O terceiro personagem é Beto, um gatinho de óculos e gravata feito em 3d com elementos 2d (FIGURA 13).

Figura 11: O garoto Ian e o urso azul Aruna.



fonte: Ian, (2019).

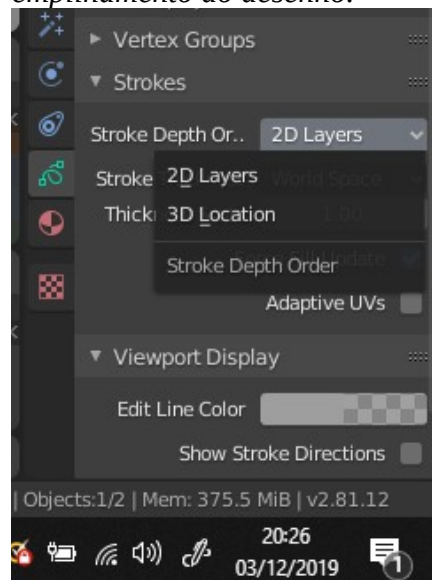
Figura 12: O gato Beto.



fonte: Ian, (2019).

Ambos personagens se encontravam em ambiente tridimensional e aproveitavam desse espaço para gerar distorções de perspectiva leves. Algo que se diferencia em fluxos de trabalhos de *softwares* bidimensionais é o posicionamento dos elementos do personagem, qual elemento fica em frente de qual. No Blender 2.8 o comportamento padrão da ferramenta *grease pencil* é seguir a ordem de empilhamento de camadas, entretanto essa ordem não pode ser animada, limitando o que pode ou não ser feito no movimento de personagens que precisam girar ou trocar de lado em tela, sendo assim necessário a troca do comportamento afim dos elementos dos personagens se encontrarem desenhados em tela de acordo com a posição tridimensional deles (FIGURA 14). Em *softwares* clássicos de animação 2d essa ordem de desenho desses elementos gráficos pode ser animada, no Blender eles tem que ser posicionados mais à frente ou mais atrás por um dos três eixos de movimentação após o ativamento da opção de posicionamento tridimensional do desenho.

Figura 13: Escolha do modo de empilhamento do desenho.



fonte: acervo pessoal, 2019.

Esse sistema traz a vantagem do personagem se encontrar diretamente no ambiente tridimensional, facilitando interações com objetos 3d, entretanto trouxe uma grande e específica desvantagem. Ao utilizar esse sistema com câmeras virtuais que tivessem distorção de perspectiva o *grease pencil* apresentava erros de desenho nas distâncias (FIGURA 15 e 16), nos forçando a separar demais as peças e gerar um processo de contra animação, cada movimento do personagem deveria ser cuidadosamente revisto para que peças se atravessando fossem reposicionadas. Esse erro não é observado com câmeras virtuais que não apresentam distorção de perspectiva, tornando esse problema algo que existe quando se quer misturar em tela o 2d de recortes e o 3d.

Figura 14: Ian com linhas aparacendo sobrepostas.



fonte: acervo pessoal, 2019.

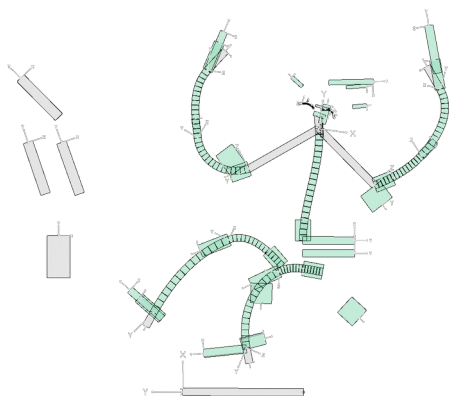
Figura 15: Grande distância entre os desenhos não impediu o comportamento.



fonte: acervo pessoal, 2019

O sistema de rigging bidimensional usa das mesmas ferramentas de rigging tridimensional do Blender, uma ou outra opção não funcionarão por necessidade de objetos tridimensionais de fato. Esse sistema trouxe a facilidade de portar todo o conhecimento prévio da ferramenta e também a utilização de sistemas automatizantes pré existentes como cinemática inversa ou física diretamente em ossos. Os personagens tiveram seus rigs feitos de maneira semelhante, foram criados para que houvesse deformação “mangueira de borracha”, suportassem mudanças de lado, trocas de partes e deformações gerais pelo corpo inteiro. O sistema de deformações a partir de ossos funciona de maneira semelhante ao 3d, a importância dada a cada osso, conhecida como *weight* ou peso, deve ser distribuída pelos pontos do desenho.

Figura 16: Esqueleto de deformação e controle de um personagem.



fonte: acervo pessoal, 2019.

Figura 17: Ian com sobreposição de seu esqueleto.



fonte: acervo pessoal, 2019.

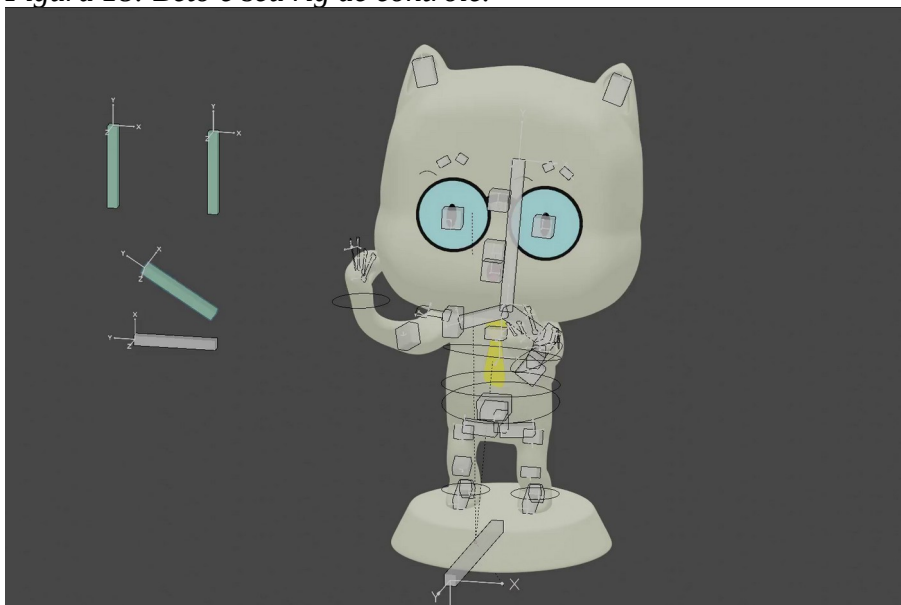
O sistema tridimensional de *rigging* traz diversas vantagens, principalmente com a possibilidade de automatizações, entretanto algumas dificuldades foram inerentes ao processo de descoberta, a ação de espelhar os personagens se torna um tanto complexa, pois durante a produção percebemos que ao fazer uma simples operação de escala negativa toda a lógica do esqueleto e mesmo a forma de selecioná-lo davam problemas, isso atrapalhou o trabalho já que não era esperado. A solução foi espelhar alguns objetos e reposicionar ossos cuidadosamente, depois construir pequenas automatizações para que o virar de personagem fosse algo simples de fazer. Portanto é interessante se ter em mente isso ao fazer um *rigging* de 2d no Blender.

Outro problema encontrado durante a produção do 2d foi com relação a performance, um dos animadores teve que reorganizar seu trabalho em torno de outro *software* quando trabalhou em casa, pois segundo o próprio animador seu computador ficou muito lento. Na época do ocorrido não conhecíamos soluções para corrigir isso, hoje sabemos que é possível melhorar a performance através de configurações no modo de exibição do objeto gráfico bidimensional, entretanto não é uma tarefa simples, envolve entrar em menus de configuração.

7 Personagem e animação 3d

No episódio piloto tivemos dois momentos de uso do 3d, no quarto do Ian, todo o cenário e objetos que o povoam são em 3d. O quarto e móveis foram modelados e os brinquedos foram baixados em sites que disponibilizam modelos em domínio público. O outro momento é o personagem Beto que tem o corpo e alguns elementos tridimensionais, mas sua boca, nariz e olhos são bidimensionais (FIGURA 19).

Figura 18: Beto e seu rig de controle.

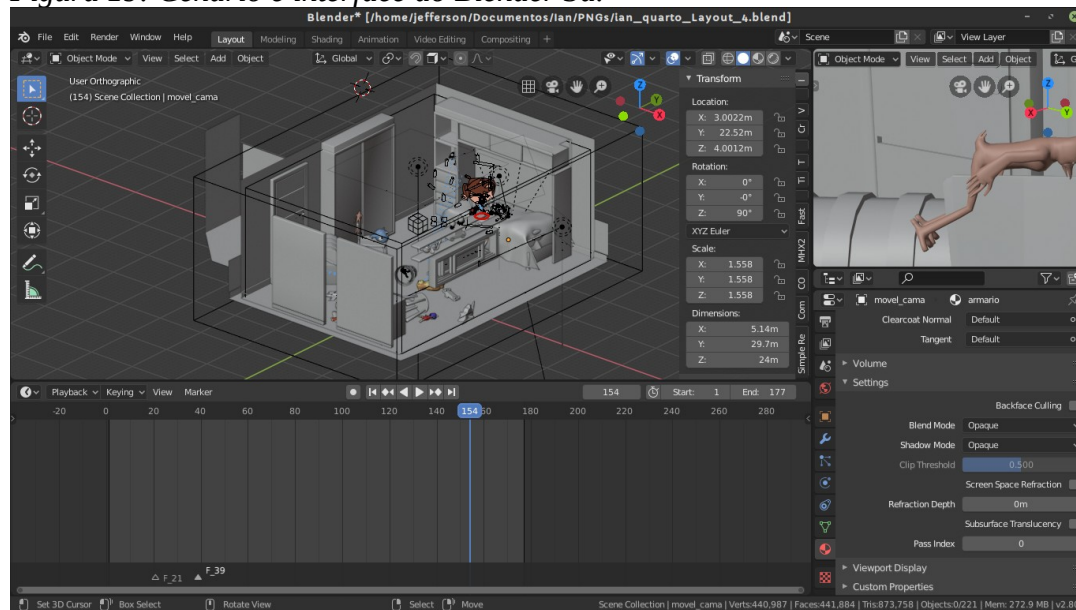


fonte: acervo pessoal, 2019.

O cenário teve o fluxo de trabalho tridimensional comum: modelagem, abertura de malhas para texturização em alguns casos e em outros métodos procedurais simples na criação dos materiais. Os brinquedos foram tratados para terem escala semelhante ao do nosso modelo e todos os materiais foram procedurais para maior agilidade com um ou outro recebendo mais atenção de acordo com sua proximidade da câmera nos planos. Física de tecidos foi utilizada para posicionar rápida e realisticamente o lençol da cama e física de corpos rígidos em alguns brinquedos espalhados pelo cenário. Todo o quarto e seus objetos, inclusive o Beto quando se encontra no quarto tiveram seus materiais tratados para ter acabamento realista. Utilizamos do motor de render *eevee*⁷ que possibilitou certo grau de realismo com agilidade de render. Na maior parte dos casos foi feito o uso do material principal, um material genérico criado para gerar resultados aceitáveis rapidamente.

⁷ Motor de renderização do Blender que usa técnicas de pré cálculo de iluminação para gerar as imagens.

Figura 19: Cenário e interface do Blender 3d.



fonte: acervo pessoal, 2019.

Figura 20: Quadro renderizado com todos os efeitos.



fonte: acervo pessoal, 2019.

Beto foi o caso mais diferente nesse trabalho, ele é um modelo 3d normal, entretanto ele tem duas variações de materiais, uma mais realista para o uso no quarto e uma cartoon para uso na imaginação de Ian. Ele também possui gráficos bidimensionais de *grease pencil* como parte essencial de seu corpo. Olhos, nariz e boca, que tiveram seu *rigging* feito junto com o corpo. Apesar

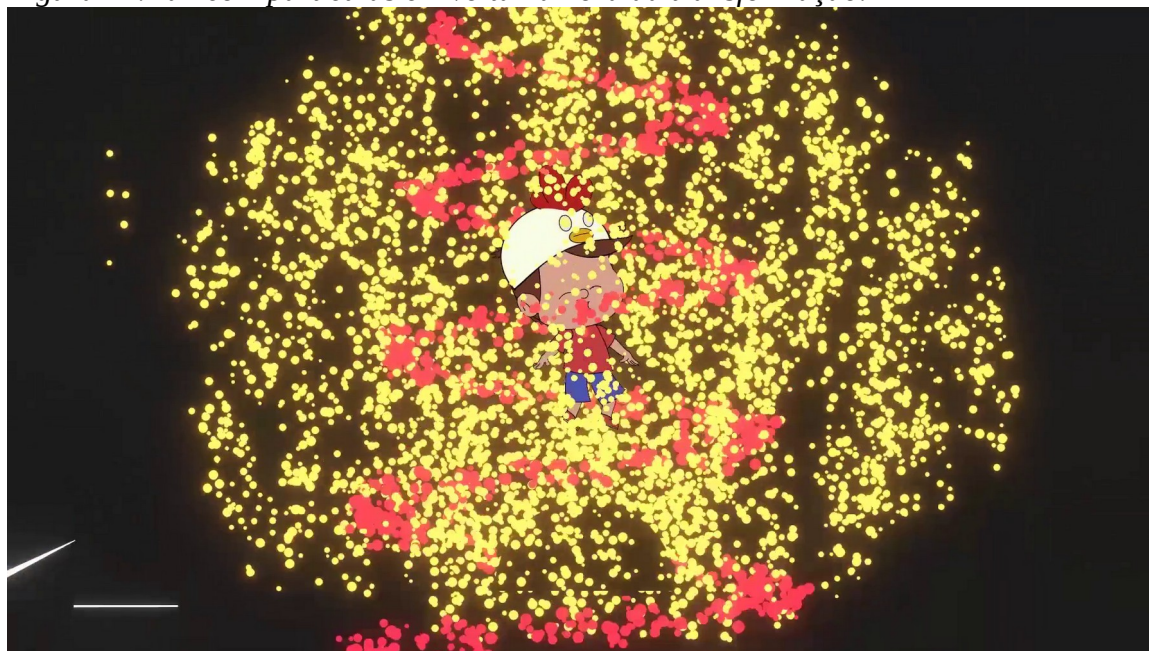
de ser um personagem híbrido que em primeira análise parece que será complicado Beto não apresentou problemas de criação e uso.

O fluxo de trabalho de 3d foi o que não teve problemas no episódio piloto, o que era para supostamente funcionar acabou funcionando, o que levou a resultados satisfatórios e esperados.

8 Partículas

Ian também teve uso de partículas, três sistemas foram utilizados, todos para a criação da cena de transição do real para o imaginário. O fluxo do sistema de partículas não teve nada diferenciado de um fluxo de trabalho normal no Blender. Primeiro se definiu com *storyboard* o que aconteceria, no segundo momento as partículas foram criadas e forças físicas foram adicionadas para dar forma ao que seria a versão final das partículas. Um sistema ao fundo formando um padrão de linhas em diagonal, dois sistemas de cores diferentes formando um turbilhão que encobre o Ian bidimensional. O mais interessante nesse caso é a interação do bidimensional e de partículas tridimensionais, ambos dividindo espaço em tela sem uso de máscaras ou outros processos relacionados à fusão de imagens em pós. Nessa mesma cena temos também o uso mais dinâmico das possibilidades da câmera em ambiente tridimensional, ela rodopia, segue o chapéu, se aproxima e afasta de Ian e das partículas.

Figura 21: Ian com partículas em volta na hora da transformação.



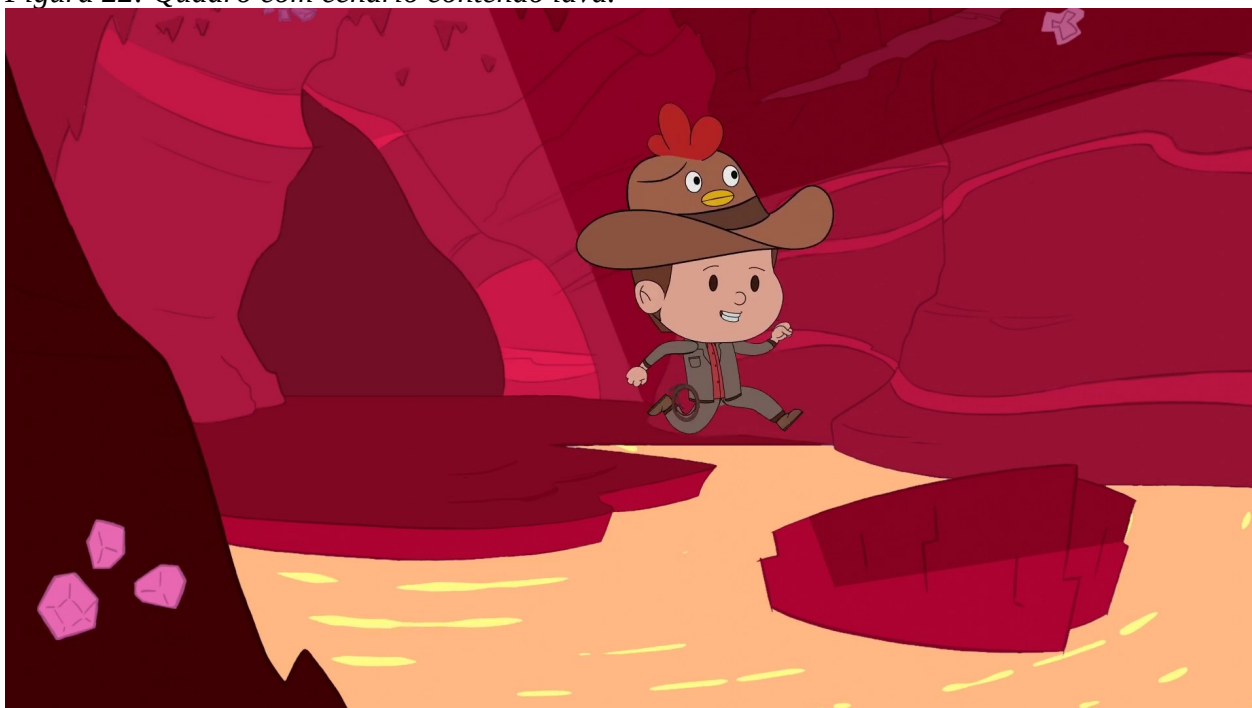
fonte: acervo pessoal, 2019.

9 Animação Paramétrica

Esta técnica de animação é voltada diretamente ao reuso e geração de muito conteúdo com pouco esforço. Animações paramétricas são todas aquelas cujo resultado se dá através das configurações de parâmetros definidos, por exemplo a velocidade. Um sistema de animação por parâmetros muito utilizado é a animação de partículas, onde definimos valores como gravidade e aceleração angular para gerar rapidamente diferentes resultados.

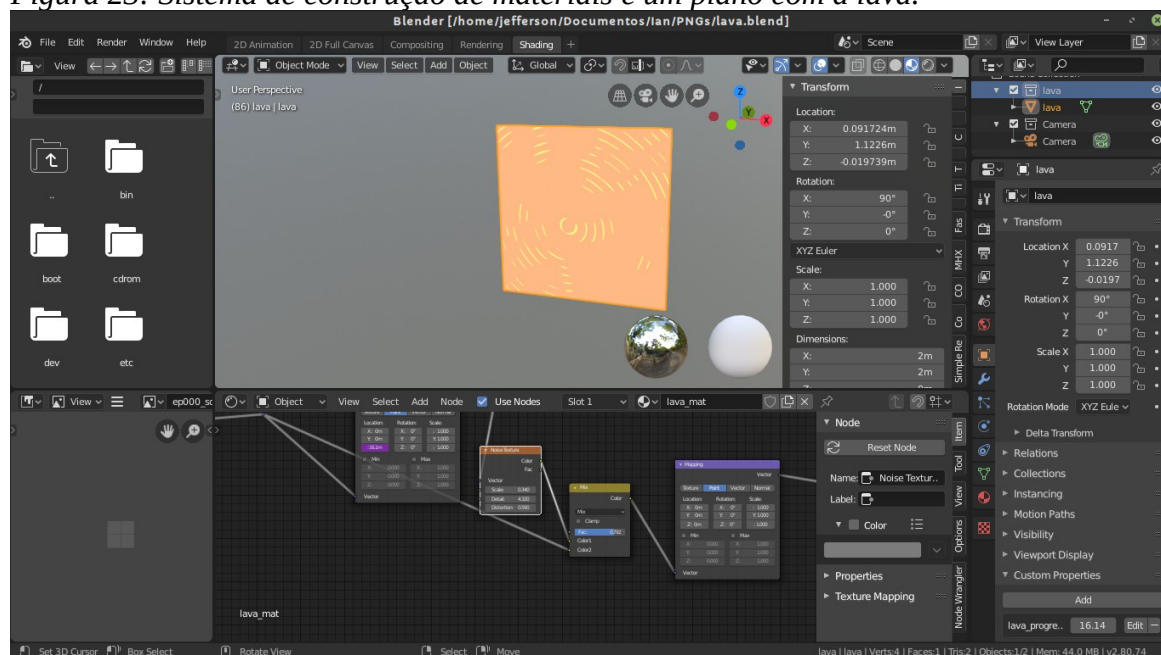
Na imaginação de Ian existe lava, todo o efeito de lava (FIGURA 23 e 24) foi feito através das ferramentas de edição de materiais do Blender, o autor desenvolveu com o uso de texturas geradas por computador e alguns cálculos vetoriais o que a direção de arte requisitou do efeito. A animação foi adicionada transformando grandezas numéricas nesses cálculos. Isso permitiu a edição rápida desse efeito para que se adequasse a qualquer momento em que aparecia. Isso evitou que tivéssemos que animar mais de uma vez a lava para que se adequasse ao desejado. Essa etapa também não teve problemas, foi mais rápida do que se imaginava e o resultado satisfaz o requerido.

Figura 22: Quadro com cenário contendo lava.



fonte: acervo pessoal, 2019.

Figura 23: Sistema de construção de materiais e um plano com a lava.



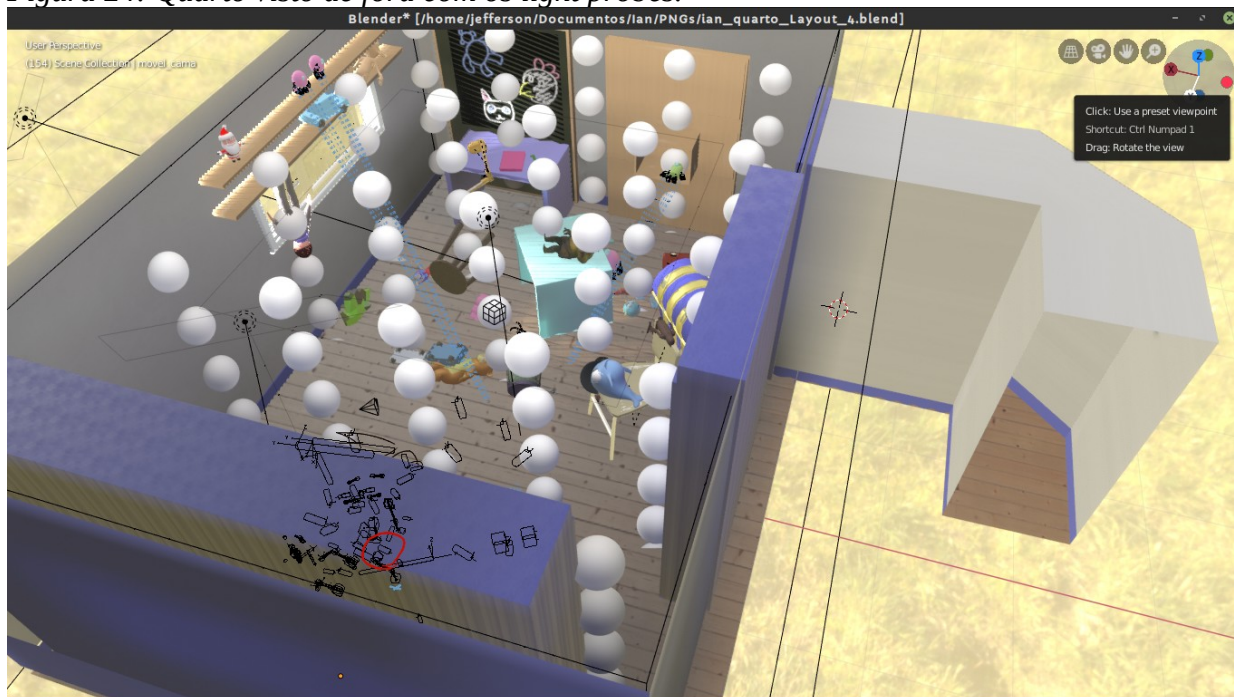
fonte: acervo pessoal, 2019.

10 Render

O processo de render foi feito de duas maneiras, usando o mesmo motor de renderização, *eevee*, que utiliza tecnologias recentes de motores gráficos de jogos para agilizar esse processo que pode levar minutos, horas ou dias a poucos segundos. Nas partes bidimensionais, na imaginação de Ian, uma câmera virtual sem distorções de perspectiva, ortográfica, foi utilizada. Os cenários previamente pintados eram imagens em planos tridimensionais posicionados em diferentes distâncias e a imagem final era gerada sem maiores ajustes.

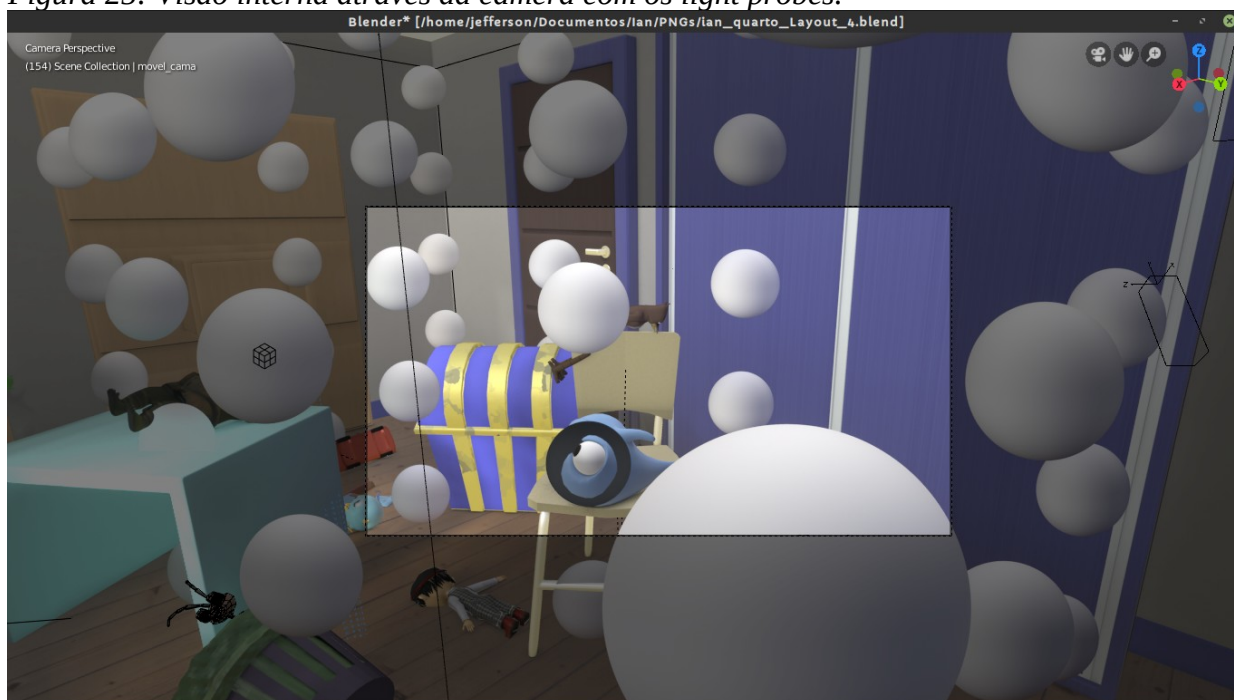
Os planos do quarto foram mais trabalhosos, a iluminação foi construída e técnicas de iluminação global foram utilizadas para alcançar maior realismo. A técnica de iluminação global utilizada foi o de cúpulas de pré iluminação, os *light probe* ou sondas de luz, sua função é pré calcular como a luz se comporta no mundo real e deixar essa informação gravada nos próprios objetos (FIGURA 25 e 26), essa técnica economiza poder computacional e com isso tempo. No Blender ela é relativamente simples, usamos um *light probe* em grelha de 5x5x5 cobrindo todo o volume do quarto e ativamos o pré cálculo. As luzes principais serviram como uma simulação do Sol e algumas de preenchimento foram adicionadas para ajudar a iluminar de maneira satisfatória algumas partes mais escondidas do quarto. No quarto utilizamos câmeras com distorção de perspectiva e foram feitos movimentos e trocas de ângulos para usar todo o potencial que podíamos da mídia.

Figura 24: Quarto visto de fora com os light probes.



fonte: acervo pessoal, 2019.

Figura 25: Visão interna através da câmera com os light probes.



fonte: acervo pessoal, 2019.

No render alguns problemas também foram observados, ao usar o *eevee* existiam, na época, problemas inerentes de iluminação. Estes tiveram que ser corrigidos com uso de geometria extra, o problema era o vazamento de luz em cantos do quarto. A solução foi criar planos por fora do quarto que cobrissem esses pontos de vazamento.

11 Considerações finais

Ao olharmos o que enfrentamos ao longo do desenvolvimento do curta é perceptível que a maioria dos problemas foram gerados com a ferramenta bidimensional. Dois problemas foram os maiores ao meu ver, sendo o primeiro deles o comportamento errático da ferramenta quando exposta a visualização com distorção de perspectiva, fato que dificulta o uso da mesma com animação de recortes integradas ao ambiente tridimensional. Esse problema não foi observado quando trabalhamos sem a distorção. O outro problema maior é a performance. Computadores com *hardware* mais simples ou desatualizados sofrem com problemas de lentidão, as configurações de diminuição de qualidade ajudam muito, entretanto são escondidas no painel de configurações completas do *software*.

Deparamo-nos com pontos do nosso fluxo de trabalho que poderiam ter sido realizados diretamente no Blender e não foram. As principais áreas que podem ter seus fluxos de trabalho alterados são a criação de peças promocionais, ilustração de cenários e os *motion graphics* diversos necessários. A vantagem de se criar as peças promocionais direto na ferramenta é a possibilidade de reuso em episódios posteriores, devido a natureza imaginativa e de brincadeira da série as peças promocionais foram criadas de maneira livre e geraram ilustrações condizentes com possíveis episódios. Essas ilustrações ou pedaços delas poderiam ser incorporadas livremente à série diminuindo o tempo de produção futuro.

A ilustração de cenários seria outro ponto chave de ser feito diretamente no *software*, a integração do cenário geraria possibilidades de organização e reutilização de elementos para montagem dos cenários sem a necessidade de passar por um processo de exportação prévia dos elementos em diversos arquivos. Essa criação de elementos seria integrada a uma biblioteca para uso posterior.

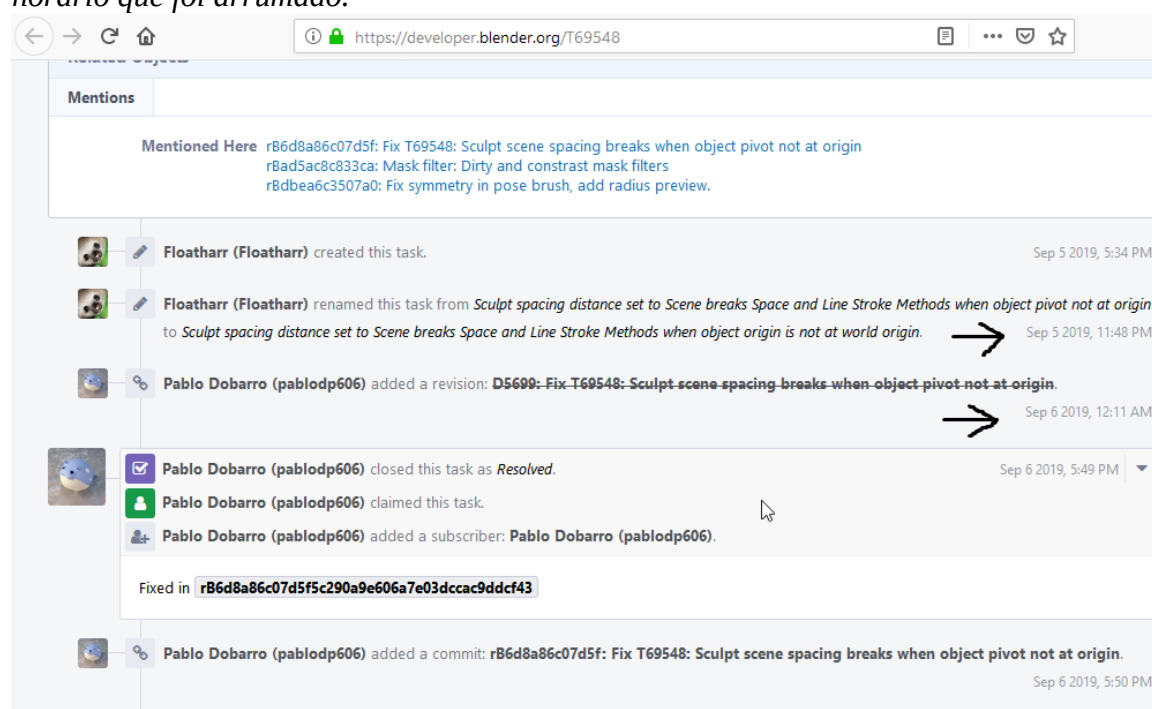
A criação de *motion graphics* foi parcialmente feita no *Blender*, A abertura do episódio foi completamente feita no Blender. Ela envolve *motion graphics* integrados com a animação dos personagens. Aqui discorro sobre a possibilidade da peça final de créditos, assim como nos outros casos, a criação dos créditos diretamente no *software* principal abre formas de interação criativa com o mesmo ou simplesmente uma tematização rápida dos créditos com os elementos pré prontos, como personagens, animações e cenários feitos para o episódio.

A partir da obra de Lévy(2009) é possível chegar também à conclusão que a criação e manutenção de *softwares open source*, ou pelo menos de alguns deles como o Blender, se encaixam no ciclo completo do movimento social da cibercultura. Com os três princípios satisfeitos,

interconexão, comunidades virtuais e inteligência coletiva. Inclusive fechando o ciclo em que Levy(2009) observa as realizações de conferências e agrupamentos como retorno ao arcaico.

Todo esse ritmo frenético do ciclo de criação então traz agilidade ao processo de criação dessa ferramenta, onde a inteligência coletiva da comunidade eleva o nível do *software* em velocidades maiores do que equipes pagas por grandes empresas (FIGURA 27). Vale ressaltar que existem sim profissionais pagos para trabalhar no Blender (FIGURA 28), contudo o histórico desses trabalhadores é de contribuição ampla e gratuita à comunidade antes mesmo de haver o convite de remuneração (FIGURA 29).

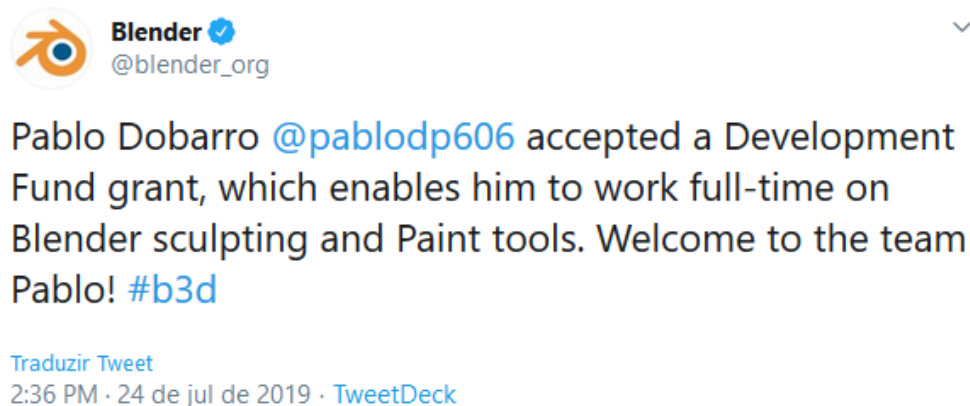
Figura 26: As setas destacam o horário da entrada do pedido de correção de bug e o horário que foi arrumado.



fonte: <https://developer.Blender.org/> acesso em: 18 de outubro de 2019

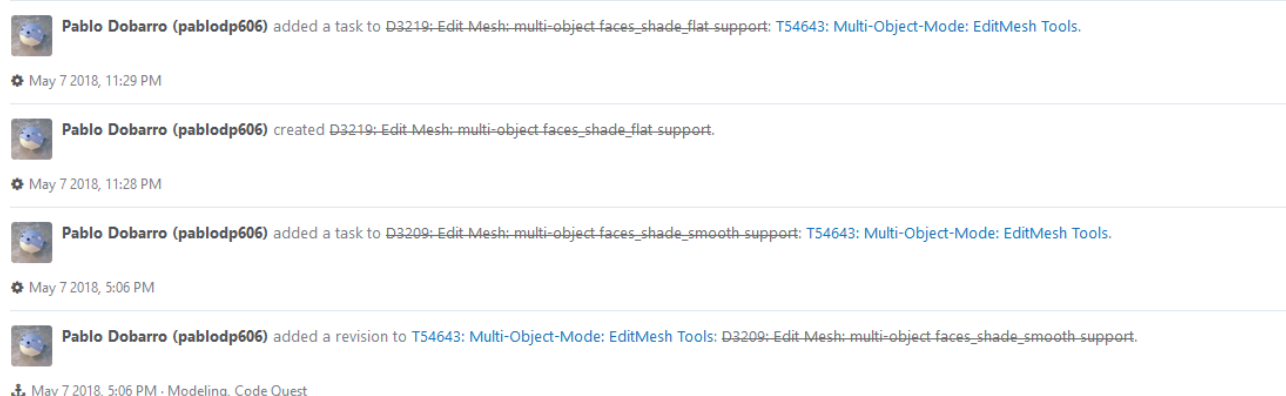
Através de reflexões sobre a obra de Flusser(2008) nota-se como há a possibilidade de empoderamento do operador através do uso crítico dessa modalidade de *software*. Havendo assim algum deslocamento do poder do programador para o artista. Em um deslocamento mais breve temos a situação do trabalho conjunto da comunidade em torno do *software*, em que artistas usam e dão respostas ao que seriam comportamentos desejados e membros codificam isso no *software*, seja no principal ou em alguma versão mais específica. O deslocamento mais pesado se dá na forma do artista híbrido, que ao se deparar com demandas de outros ou próprias dispende tempo para aprender sobre linguagens de programação e modificar o *software* em seu código fonte.

Figura 27: Twitter oficial da Blender foundation anunciando a contratação de Pablo Dobarro.



fonte: https://twitter.com/Blender_org/status/1154082816919003137
 acesso em: 18 de outubro de 2019

Figura 28: Histórico de submissões de alteração de código por Pablo Dobarro datando de antes da contratação.



fonte: <https://developer.Blender.org/> acesso em: 18 de outubro de 2019

Talvez seja interessante notar que o ambiente do Blender 3d dá ferramentas e liberdade o suficiente para que sejamos artistas híbridos em diferentes níveis, uma breve modificação de interface para que o *software* seja adequado ao seu uso e não o contrário é simples de executar, ainda que não esteja entrando no mundo da programação o artista já desenvolveu alguma criticidade sobre seu ato a ponto de a não satisfação com o dado pelo *software* gerar uma modificação. Os níveis de complexidade aumentam conforme o interesse do artista, alguns vão construir soluções através de linguagens de script, construir extensões e por fim modificar o código fonte.

Outro ponto interessante de se pensar nesse caso é o acesso à ferramenta. A gratuidade e facilidade de execução do Blender nos mais diversos cenários (computadores antigos têm acesso através de versões antigas, quem não pode instalar o programa pode usar uma versão portátil e a ampla retrocompatibilidade) se soma à dinâmica do movimento da cibercultura de Levy (2009). A

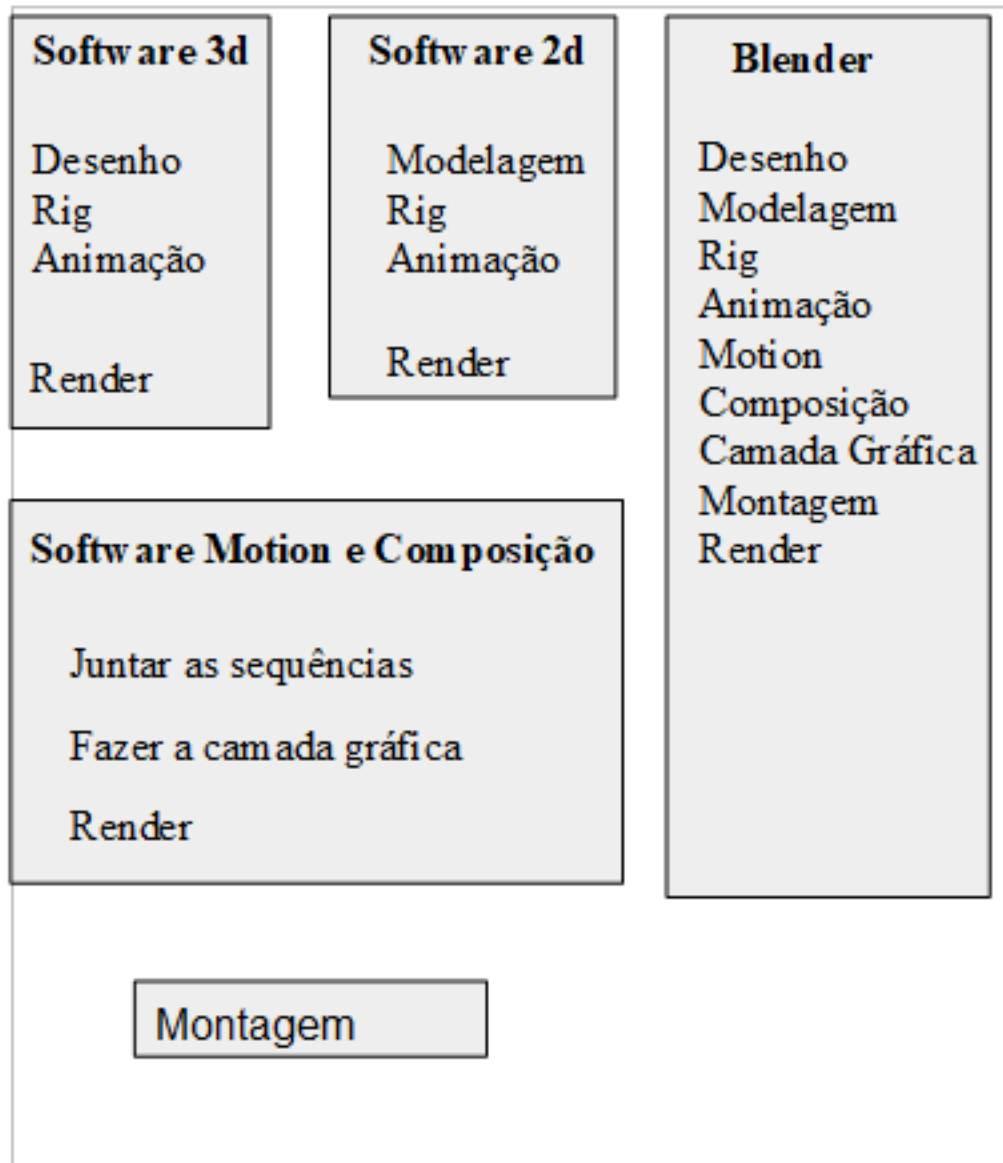
facilidade de acesso gera maior volume de criação, principalmente no momento de escrita deste artigo, mas não só discorremos de criação artística, mas educacional e de código do próprio programa também. Isso acarreta em uma comunidade crescente e um efeito de bola de neve, quanto mais se usa mais se gera material, quanto mais material mais pessoas querem usar e assim por diante. A velocidade da evolução das funções do Blender 3d é grande, logo após o término de Ian já existiam novidades e durante a finalização desse texto a versão 2.82 é preparada para lançamento com diversas melhorias e novidades, por exemplo, o suporte a pintura de vértices do *grease pencil*, que permite acabamentos pintados.

A importância de datar esse trecho se dá por apesar de existir há pelo menos duas décadas, o Blender e sua comunidade terem achado tração nos últimos tempos. Há de se especular que todo um panorama político-social e de ordem das novas dinâmicas de taxação do uso do *software* para criação artística somados a constantes atualizações e melhorias na experiência do usuário possam ter levado ao estado atual da comunidade do *software* livre de criação artística, sem ignorar todos os jovens que assim como eu aprenderam usando esse *software* e tantos outros que agora se inserem no mercado, talvez frustrados com o que lhes é imposto usar. Entretanto, isso é assunto para outro estudo.

Outro olhar que pode ser jogado em torno de suítes de criação seriam estudos das relações de trabalho modernas, um trabalhador que faz tudo, um *software* que faz tudo. Esta situação gera inclusive coisas estranhas no sentido de *software*, como a suíte adobe em que mais de um *software* faz as mesmas coisas e o próprio Blender 3d em que um *software* pode suprir toda uma linha de produção. Essas soluções podem servir para “seduzir” o trabalhador a poder fazer mais de uma coisa e com isso se achar no controle da ferramenta. Nasce como uma novidade a fim de facilitar sua expressão criativa e é cooptada pelo sistema das relações de trabalho, em que o chefe sabe que dá pra fazer ou o trabalhador revela em algum momento o que pode ser feito e daí surge o trabalhador que tem que ser multidisciplinar e trabalhar dobrado. Por esse viés mais negativo, em meu ver, podemos imaginar o por que do surgimento do artista híbrido, que ocupa mais de uma competência de campos diferentes para gerar maior valor de produção.

Contudo, não gostaria de ser cínico e apenas ver um lado maldoso no presente artigo, pois se por um lado há precarização do artista profissional através da multidisciplinaridade há também a questão da expressão do próprio artista e sua autonomia. Se o artista sabe utilizar sua ferramenta de diversas formas ele tem a possibilidade de criações diferentes, usando das interações entre os diversos campos que sua ferramenta abrange, tarefas podem ser aceleradas e ideias novas podem surgir.

Figura 29: Fluxograma simplificado de fluxo de trabalho em outros softwares e no Blender 3d.



fonte: acervo pessoal, 2019

Multitechnical animation workflow with Blender 3D

ABSTRACT

The purpose of this paper is to detail the workflow of 2D animation series using the free and open source Blender 3D tool. This study focuses on the critical analysis of animation techniques and procedures, observed through the author's practical production throughout the course and focusing on the conclusion short Ian. It also seeks to establish an overview of the development of open source 2D animation programs and how such solutions are developed by the user community. As a methodological procedure, the research adopts an empirical path of observation and analysis, supported by theoretical references related to the themes of technical images, cyberculture and multidisciplinary.

Palavras-chave: workflow, Blender 3d, 2d animation, animation series market

Referências Bibliográficas

- About* da ASWF. 2018, Disponível em: <<https://www.aswf.io/about/>>. Acesso em 20 junho, 2019.
- AMIDI, A. Studio Ghibli and ‘Futurama’ Being Made Free and Open Source. 2016, Disponível em: <<https://www.cartoonbrew.com/tech/toonz-software-used-studio-ghibli-futurama-made-free-open-source-138111.html>>. Acesso em 27 maio, 2019.
- Blender and Next Gen: a Netflix Original, Disponível em: <<https://youtu.be/iZn3kCsw5D8>>. Acesso em 20 junho, 2019.
- FLUSSER, V.; Filosofia da Caixa Preta. *São Paulo: Hucitec*, 1985.
- FLUSSER, V.; O mundo codificado-por uma filosofia do design e da comunicação. São Paulo: Cosac Naify, 2013. __. O mundo das imagens técnicas—elogio da superficialidade. São Paulo: Annablume, 2008. p. 51 – 65.
- Jawadekar, W.S.; *software Enggining: Principles and Pratices*. Nova Deli: Tata McGraw-Hill Education, 2004. p. 4
- LEVY, P.; Ciberultura. São Paulo: Editora 34. 2009
- STALLMAN, R.; Por que o Código Aberto não compartilha dos objetivos do *software* Livre. s.d. Disponível em: <<http://www.gnu.org/philosophy/open-source-misses-the-point.pt-br.html>>. Acesso em: 27 maio, 2019.
- Página de *Releases* do Blender. s.d. Disponível em: <<https://www.Blender.org/download/releases/2-73/>>. Acesso em 27 maio, 2019.

Filmografia

313 Experimentos de Gerson, Direção de Emanuelle Schiavon, Pelotas: UFPEL, 2018. (3:36 min.)

A Inveja de Cléber e Um Cigarro no Banheiro, Direção de Jefferson Nascimento, Pelotas: UFPEL, 2017. (2:36 min.)

Elephants Dream, Direção de Bassam Kurdali, Holanda: Media Art Institute, 2005. (10:54 min)

Hero, Direção de Daniel Martinez Lara, Holanda: Blender Foundation, 2018. (3 min.)

Ian, Direção de Jefferson Nascimento. Pelotas: UFPEL, 2019. (3:22 min.)

Incrível Mundo de Gumball [Seriado], Direção: Mic Graves, Antoine Peres. Produção: Ellen Collins, Sarah Fell, Daniel Lennard. Reino Unido: Cartoon Network Development Studio Europe, 2018. (11 min.).

Jerry, O Coala, Direção de Jefferson Nascimento, Pelotas: UFPEL, 2016. (4:39 min)

Next Gen, Direção de Kevin R. Adams e Joe Ksander. China: Baozou Manhwa, Alibaba Pictures , Canadá: Tangent Animation, 2018. (105 min.)

The Kingdom of Dreams and Madness, Direção de Mami Sunada. Japão: Dwango, 2013. (118 min.)

Índice de figuras

Figura 1: Interface do opentoonz com um arquivo do autor aberto.fonte: acervo pessoal.....	8
Figura 2: Quadro do curta Jerry, O coala.fonte: Jerry, O Coala (2016).....	8
Figura 3: Interface do Synfig.fonte: acervo pessoal.....	9
Figura 4: Cena do videoclipe 313 experimentos de Gerson na Interface do Blender 2.79b com a extensão COA Tools.fonte: acervo pessoal.....	10
Figura 5: Quadro do curta A Inveja de Cléber e um cigarro no banheiro. O detalhe do fogo do isqueiro é feito em 2d em espaço tridimensional com <i>grease pencil</i> .fonte: A inveja de Cléber e um cigarro no banheiro (2017).....	10
Figura 6: Quadro do animatic do curta, desenhando em espaço tridimensional.....	11
Figura 7: Cartaz de divulgação dos projetos opensource apoiados pela AMPAS.fonte: https://www.aswf.io/divulgacao	12
Figura 8: Interface montada para desenho e manipulação de keyframes de desenho e interpolação.fonte: acervo pessoal, 2019.....	16
Figura 9: Interface para montagem utilizada para fazer o animatic.fonte: acervo pessoal, 2019.....	16
Figura 10: Na esquerda o vetor gerado pelo grease pencil e na direita um vetor gerado em um <i>software</i> de edição de vetores comum.....	18
Figura 11: O garoto Ian e o urso azul Aruna.fonte: Ian, (2019).....	18
Figura 12: O gato Beto.fonte: Ian, (2019).....	19
Figura 13: Escolha do modo de empilhamento do desenho.fonte: acervo pessoal, 2019.....	19
Figura 14: Ian com linhas aparacendo sobrepostas.fonte: acervo pessoal, 2019.....	20
Figura 15: Grande distância entre os desenhos não impediu o comportamento.fonte: acervo pessoal, 2019.....	20
Figura 16: Esqueleto de deformação e controle de um personagem.fonte: acervo pessoal, 2019.....	21
Figura 17: Ian com sobreposição de seu esqueleto.fonte: acervo pessoal, 2019.....	21
Figura 18: Beto e seu rig de controle.fonte: acervo pessoal, 2019.....	23
Figura 19: Cenário e interface do Blender 3d.fonte: acervo pessoal, 2019.....	24
Figura 20: Quadro renderizado com todos os efeitos.fonte: acervo pessoal, 2019.....	24
Figura 21: Ian com partículas em volta na hora da transformação. fonte: acervo pessoal, 2019.....	25
Figura 22: Quadro com cenário contendo lava.fonte: acervo pessoal, 2019.....	26
Figura 23: Sistema de construção de materiais e um plano com a lava.fonte: acervo pessoal, 2019.....	27
Figura 24: Quarto visto de fora com os light probes.fonte: acervo pessoal, 2019.....	28
Figura 25: Visão interna através da câmera com os light probes.fonte: acervo pessoal, 2019.....	28
Figura 26: As setas destacam o horário da entrada do pedido de correção de bug e o horário que foi arrumado.fonte: https://developer.Blender.org/ acesso em: 18 de outubro de 2019.....	30
Figura 27: Twitter oficial da Blender foundation anunciando a contratação de Pablo Dobarro.fonte: https://twitter.com/Blender_org/status/1154082816919003137 acesso em: 18 de outubro de 2019.....	31
Figura 28: Histórico de submissões de alteração de código por Pablo Dobarro datando de antes da contratação.fonte: https://developer.Blender.org/ acesso em: 18 de outubro de 2019.....	31
Figura 29: Fluxograma simplificado de fluxo de trabalho em outros softwares e no <i>Blender 3d</i> .fonte: acervo pessoal, 2019.....	33

Anexos

NDER 3D ANIMAÇÃO MULTITÉCNICA - BLENDER 3D
BLENDER 3D ANIMAÇÃO MULTITÉCNICA - BLENDER
ITÉCNICA - BLENDER 3D ANIMAÇÃO MULTITÉCNICA

